K 6

```
FFFFFFFFFFFFFFF     000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFF              000         000  RRR         RRR  RRR         RRR        TTT        LLL
FFF              000         000  RRR         RRR  RRR         RRR        TTT        LLL
FFF              000         000  RRR         RRR  RRR         RRR        TTT        LLL
FFF              000         000  RRR         RRR  RRR         RRR        TTT        LLL
FFF              000         000  RRR         RRR  RRR         RRR        TTT        LLL
FFFFFFFFFFFF     000         000  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFFFFFFFFFFF     000         000  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFFFFFFFFFFF     000         000  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFF              000         000  RRR   RRR        RRR   RRR              TTT        LLL
FFF              000         000  RRR   RRR        RRR   RRR              TTT        LLL
FFF              000         000  RRR   RRR        RRR   RRR              TTT        LLL
FFF              000         000  RRR      RRR     RRR      RRR           TTT        LLL
FFF              000         000  RRR      RRR     RRR      RRR           TTT        LLL
FFF              000         000  RRR      RRR     RRR      RRR           TTT        LLL
FFF                 000000000     RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
FFF                 000000000     RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
FFF                 000000000     RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
**FILE**ID**FORFMTINT
```

```
FFFFFFFFFF    000000    RRRRRRRR    FFFFFFFFFF  MM      MM  TTTTTTTTTT    IIIIII    NN      NN  TTTTTTTTTT
FFFFFFFFFF    000000    RRRRRRRR    FFFFFFFFFF  MM      MM  TTTTTTTTTT    IIIIII    NN      NN  TTTTTTTTTT
FF          00      00  RR      RR  FF          MMMM  MMMM      TT          II      NN      NN      TT
FF          00      00  RR      RR  FF          MMMM  MMMM      TT          II      NN      NN      TT
FF          00      00  RR      RR  FF          MM  MM  MM      TT          II      NNNN    NN      TT
FF          00      00  RR      RR  FF          MM  MM  MM      TT          II      NNNN    NN      TT
FFFFFFFF    00      00  RRRRRRRR    FFFFFFFF    MM      MM      TT          II      NN  NN  NN      TT
FFFFFFFF    00      00  RRRRRRRR    FFFFFFFF    MM      MM      TT          II      NN  NN  NN      TT
FF          00      00  RR  RR      FF          MM      MM      TT          II      NN    NNNN      TT
FF          00      00  RR  RR      FF          MM      MM      TT          II      NN    NNNN      TT
FF          00      00  RR    RR    FF          MM      MM      TT          II      NN      NN      TT            ....
FF          00      00  RR    RR    FF          MM      MM      TT          II      NN      NN      TT            ....
FF            000000    RR      RR  FF          MM      MM      TT        IIIIII    NN      NN      TT            ....
FF            000000    RR      RR  FF          MM      MM      TT        IIIIII    NN      NN      TT            ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE FOR$$FMT_INTRP (%TITLE'Fortran Format Statement Interpreter'
0                 IDENT = '2-037'          ! File: FORFMTINT.B32  Edit: SBL2037
0                 ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 ! FACILITY: FORTRAN
1 !
1 ! ABSTRACT:
1 !
1 !       This module interprets FORTRAN format statements
1 !       which have been pre-compiled into an encoded form by either the
1 !       FORTRAN compiler or the run-time format compiler,
1 !       FOR$$FMT_COMPIL.  It is independent of READ and WRITE semantics
1 !       and is executed at both the READ Formatted and WRITE Formatted
1 !       User Data Formatters (UDF) level of abstraction.
1 !
1 ! AUTHOR: Peter Yuo, CREATION DATE: 17-Feb-77
1 !
1 ! MODIFIED BY:
1 !       Peter Yuo, 25-Feb-77, Version 1
1 ! 01    -       Original
1 !
1 !       Richard Grove, 19-Aug-77, Version 2
1 ! [Previous edit history removed.  SBL 23-Aug-1982]
1 ! 2-032 - Add defaults for O and Z format width when value is not 1, 2, 4, 8 or
1 !         16 bytes.  SBL 29-Dec-1980
1 ! 2-033- Improved fix for 2-032, courtesy of Joel CLinkenbeard.  SBL 8-Jan-1981
1 ! 2-034 - Convert FOR$$FMT_INTRP1 to JSB linkage for better performance.
1 !         JAW 29-Jul-1981
1 ! 2-035 - Miscellaneous performance enhancements:  JAW 29-Jul-1981
1 !         Check for certain specific one-byte format codes at the outset
1 !           and special-case them.
1 !         For all format codes, if optional second byte is not present,
;   1 !               bypass checks for VFEs and for optional forms of RC and W.
;   1 !             Break FI_ACT into two tables, each having 1-byte entries,
;   1 !               placing the special action in FI_ACT_2 and indicating the
;   1 !               need for special action with the low-order bit of FI_ACT.
;   1 !               Select a special action only if this bit is set.
;   1 !             For codes _DF through _DD, check for element size of 4 first.
;   1 !             Narrow the scope of ACT, FMT_REPR and P, which are not needed
;   1 !               in the outermost block, to conserve registers.
;   1 !             Replace CASE on V_RC_TYPE with IF ... THEN to avoid an EXTZV.
;   1 ! 2-036 - Correct range check of P value from VFEs.  SBL 23-Aug-1982
;   1 ! 2-037 - Allow zero-value VFEs for W, D and E fields only.  Use prologue
;   1 !             file.  SBL 26-Apr-1983
;   1 !-
;    72          0071   1 !
;    73          0072   1 ! PROLOGUE FILE:
;    74          0073   1 !
;    75          0074   1
;    76          0075   1 REQUIRE 'RTLIN:FORPROLOG';                           ! FORTRAN Definitions
;    77          0141   1 SWITCHES ZIP;                                        ! Optimize for speed
;    78          0142   1
;    79          0143   1 !
;    80          0144   1 ! TABLE OF CONTENTS:
;    81          0145   1 !
;    82          0146   1
;    83          0147   1 FORWARD ROUTINE
;    84          0148   1     FOR$$FMT_INTRP0 : JSB_FMT0 NOVALUE,              ! initialization
;    85          0149   1     FOR$$FMT_INTRP1 : JSB_FMT1 NOVALUE;              ! Interpret until a data format code
;    86          0150   1
;    87          0151   1 !
;    88          0152   1 ! MACROS:
;    89          0153   1 !
;    90          0154   1 !      NONE
;    91          0155   1 !
;    92          0156   1 ! EQUATED SYMBOLS:
;    93          0157   1 !
;    94          0158   1 !      NONE
;    95          0159   1 !
;    96          0160   1 ! OWN STORAGE:
;    97          0161   1 !
;    98          0162   1 !      NONE
;    99          0163   1 !
;   100          0164   1 !
;   101          0165   1 ! EXTERNAL REFERENCES:
;   102          0166   1 !
;   103          0167   1
;   104          0168   1 EXTERNAL ROUTINE
;   105          0169   1     FOR$$SIGNAL_STO : NOVALUE,                       ! Signal_stop FOR$_abcmnoxyz, given
;   106          0170   1                                                      ! (short) Fortran error number (FOR$K_abcmnoxyz)
;   107          0171   1                                                      ! as a parameter
;   108          0172   1     FOR$$SIGNAL : NOVALUE;                           ! Signal FOR$_abcmnoxyz, given (short)
;   109          0173   1
;   110          0174   1                                                      ! FORTRAN error number (FOR$K_abcmnoxyz)
;   111          0175   1                                                      ! as a parameter.
;   112          0176   1
```

FOR
2-0

```
1 GLOBAL ROUTINE FOR$$FMT_INTRP0                              ! Format interpreter initialization
1     : JSB_FMT0 NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Initializes the format interpreter
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       CCB                     Contains adr. of current LUB/ISB/RAB.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       CCB [ISB$W_FMT_REP]     Set repeat count to 0 to indicate no repeat
1 !                               for this statement yet.
1 !       CCB [ISB$B_FMT_P]       Set P scale factor to 0 for this statement
1 !       CCB [ISB$A_FMT_PTR]     Initializes format pointer to
1 !                               beginning
1 !       CCB [ISB$W_FMT_REVER]   Offset of current format reversion
1 !                               point
1 !       CCB [ISB$B_FMT_DEP]     Depth of repeat group pushdown stack
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2         EXTERNAL REGISTER
2             CCB : REF $FOR$CCB_DECL;
2
2     !+
2     ! Set repeat count to 0 to indicate no repeat for this statement.
2     !-
2
2     CCB [ISB$W_FMT_REP] = 0;
2
2     !+
2     ! Set P scale factor to 0 for this statement (no scaling).
2     !-
2
2     CCB [ISB$B_FMT_P] = 0;
2
2     !+
2     ! Set format flags to zero for this statement.
2     !-
2
2     CCB [ISB$W_FMT_FLAGS] = 0;
2
2     !+
2     ! Set BN flag if LUB$V_NULLBLNK is set
2     !-
2
```

```
;  171          0234  2      CCB [ISB$V_BN] = .CCB [LUB$V_NULLBLNK];
;  172          0235  2
;  173          0236  2      !+
;  174          0237  2      ! Set current format position to beginning of format.
;  175          0238  2      !-
;  176          0239  2
;  177          0240  2      CCB [ISB$A_FMT_PTR] = .CCB [ISB$A_FMT_BEG];
;  178          0241  2
;  179          0242  2      !+
;  180          0243  2      ! Initialize format reversion point to beginning of format
;  181          0244  2      ! byte array. The reversion point is used when there are
;  182          0245  2      ! more user data elements than data format codes.
;  183          0246  2      ! Since it is a 16-bit offset with respect to ISB$A_FMT_BEG, set to 0.
;  184          0247  2      !-
;  185          0248  2
;  186          0249  2      CCB [ISB$W_FMT_REVER] = 0;
;  187          0250  2
;  188          0251  2      !+
;  189          0252  2      ! Initialize format repeat group push down stack depth
;  190          0253  2      ! to empty (-1).  0 = 1 item, 1 = 2 items in stack, etc.
;  191          0254  2      !-
;  192          0255  2
;  193          0256  2      CCB [ISB$B_FMT_DEP] = -1;
;  194          0257  2
;  195          0258  2      !+
;  196          0259  2      ! Initialize ISB$B_FMT_CODE to zero, which will tell
;  197          0260  2      ! FOR$$UDF_WF9 not to call FOR$$UDF_WF1 unless there were no
;  198          0261  2      ! items in the I/O list.
;  199          0262  2      !-
;  200          0263  2
;  201          0264  2      CCB [ISB$B_FMT_CODE] = 0;
;  202          0265  2
;  203          0266  2      !+
;  204          0267  2      ! All other ISB locations and flags have already been
;  205          0268  2      ! initialized to 0 or a specified value by the I/O statement
;  206          0269  2      ! initialization for this I/O statement.
;  207          0270  2      !-
;  208          0271  2
;  209          0272  2      RETURN;
;  210          0273  1      END;                                   ! End of routine FOR$$FMT_INTRP0

                                                                 .TITLE  FOR$$FMT_INTRP Fortran Format Statement Interpr
;                                                                                       eter
                                                                 .IDENT  \2-037\

                                                                 .EXTRN  FOR$$SIGNAL_STO
                                                                 .EXTRN  FOR$$SIGNAL

                                                                 .PSECT  _FOR$CODE,NOWRT,  SHR,  PIC,2

                                            8D  AB  B4 00000 FOR$$FMT_INTRP0::
                                                                 CLRW    -115(CCB)                          ; 0216
                                            88  AB  94 00003     CLRB    -120(CCB)                          ; 0222
                                            93  AB  B4 00006     CLRW    -109(CCB)                          ; 0228
         50       FF   AB             01        06  EF 00009     EXTZV   #6, #1, -1(CCB), R0                ; 0234
     93  AB            01             00        50  F0 0000F     INSV    R0, #0, #1, -109(CCB)              ;
```

```
                    80   AB      FF7C   CB  D0 00015          MOVL    -132(CCB), -128(CCB)                     ; 0240
                                   90   AB  B4 0001B          CLRW    -112(CCB)                                ; 0249
                    92   AB             01  8E 0001E          MNEGB   #1, -110(CCB)                            ; 0256
                                   8F   AB  94 00022          CLRB    -113(CCB)                                ; 0264
                                            05 00025          RSB                                              ; 0273

; Routine Size:  38 bytes,    Routine Base:  _FOR$CODE + 0000
```

```
GLOBAL ROUTINE FOR$$FMT_INTRP1              ! Format interpret until data code
!         (EL_SIZE,                         ! Size in addressable units of data elements
                                            ! This argument is passed in
                                            ! external register EL_SIZE.
!         DT_SEEN)                          ! 1 if data transmitter seen
                                            ! This argument is passed in
                                            ! external register DT_SEEN.
    : JSB_FMT1 NOVALUE =                    ! Value is format code
                                            ! Returned in external register
                                            ! FMT_CODE.

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       FOR$$FMT_INTRP1 interprets FORTRAN format statements
!       which have been precompiled into an encoded form by either
!       the FORTRAN compiler or the run-time format compiler,
!       FOR$FMT_COMPIL. Only FOR$$FMT_INTRP1 understands the structure
!       and encoding of compiled format statements. Furthermore,
!       it is independent of READ and WRITE semantics.
!       Each call to FOR$$FMT_INTRP1 processes as many format
!       codes as possible until it encounters one which
!       needs to access user program data, needs to access the
!       data buffer, or depend on whether read or write. This
!       block is independent of whether a READ or WRITE is
!       being performed. It is invoked in both the formatted
!       READ and WRITE user data formatter routines (UDF).
!
!       Note: being compatible with -11 OTS, there is a
!       difference between nf and n(f) in that VFE's are evaluated only
!       once for the former and each time for the latter. The former
!       is termed, repeating a format code while the latter is
!       termed a repeat group.
!
! IMPLICIT INPUTS:
!
!       EL_SIZE                 Size in addressable units of user
!                               data element. Used to set default
!                               widths (W) for default format.
!                               A value of 0 indicates that this is
!                               the end of the I/O list call and there is
!                               no user I/O list element to be transmitted.
!
!       DT_SEEN                 1 if a data transmitter has been seen by
!                               the current call to the UDF level, 0
!                               otherwise.  If it is set, we don't evaluate
!                               format items for data transmitters.
!
!       OTS$$A_CUR_LUB          Adr. of current logical unit block
!                               (LUB) Used to setup base (ISB) to
!                               I/O statement block.
!
!    The following locations are set only by previous calls to
!    FOR$$FMT_INTRP1, i.e., are effectively OWN:
!
!       CCB [ISB$A_FMT_PTR]     Adr. of next byte in compiled
```

```
1 !                                       format byte array. A value of 0
1 !                                       indicates that this is the end of the
1 !                                       I/O list call and there is no user
1 !                                       I/O list element to be transmitted.
1 !         CCB [ISB$V_USER_ELEM]         0 until a user element format
1 !                                       code seen. Infinite loop preventer
1 !         CCB [ISB$W_FMT_REP]           Current format code repeat count (n)
1 !                                       or 0 if not repeating a single
1 !                                       format code. Note: the repeat
1 !                                       count for a repeat group is kept
1 !                                       in the top of the format stack, not here.
1 !
1 !         CCB [ISB$A_FMT_BEG]           Adr. of beginning of format statement
1 !         CCB [ISB$B_FMT_DEP]           Depth of repeat group format pushdown stack.
1 !         CCB [ISB$W_FMT_STKP]          Stack of offsets to beginning of repeat groups
1 !         CCB [ISB$W_FMT_STKR]          Stack of group repeat counts
1 !         CCB [ISB$W_FMT_REVER]         Offset of current format reversion
1 !                                       point to revert to when end of format
1 !                                       statement is encountered with more data
1 !                                       elements to be transmitted.
1 !         CCB [ISB$V_USER_ELEM]         Flag: 1 if seen a user data element format code,
1 !                                       0 if not.  Used to check for infinite format loop
1 !                                       in which no user data element format codes are present
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !     The following are outputs only to a successive call to
1 !     FOR$$FMT_INTRP(0,1), i.e., are effectively OWN.
1 !
1 !         CCB [ISB$V_USER_ELEM]         0 if no user data element format
1 !                                       code seen this repeat group, 1
1 !                                       if one or more
1 !         CCB [ISB$W_FMT_REP]           Current format code repeat count (n)
1 !                                       or 0 if not repeating a single
1 !                                       format code. Note: the repeat
1 !                                       count for a repeat group is kept
1 !                                       in the top of the format stack, not here.
1 !         CCB [ISB$B_FMT_DEP]           Depth of repeat group format pushdown stack.
1 !         CCB [ISB$W_FMT_STKP]          Stack of offsets to beginning of repeat groups
1 !         CCB [ISB$W_FMT_STKR]          Stack of group repeat counts
1 !         CCB [ISB$W_FMT_REVER]         Offset of current format reversion
1 !                                       point to revert to when end of format
1 !                                       statement is encountered with more data
1 !                                       elements to be transmitted.
1 !
1 !     The following are output to available to the caller (read
1 !     or write user data formatter):
1 !
1 !         CCB [ISB$A_FMT_PTR]           Adr. of next byte to be read from
1 !                                       the compiled format statement byte array
1 !                                       are pushed as a pair.
1 !         CCB [ISB$B_FMT_P]             Signed scale factor (P)
1 !         CCB [ISB$W_FMT_W]             Unsigned width of field (W)
1 !         CCB [ISB$B_FMT_D]             Unsigned number of digits in fraction (D)
1 !         CCB [ISB$B_FMT_E]             Unsigned number of characters
1 !                                       in exponent (E).
1 !         CCB [ISB$V_USER_ELEM]         Flag: 1 if seen a user data element format code,
;   326          0388  1 !                                  0 if not.  Used to check for infinite format loop
;   327          0389  1 !                                  in which no user data element format codes are present
;   328          0390  1 !
;   329          0391  1 ! SIDE EFFECTS:
;   330          0392  1 !
;   331          0393  1 !        SIGNAL_STOPs FOR$_SYNERRFOR (62='SYNTAX ERROR IN FORMAT'')
;   332          0394  1 !        SIGNAL_STOPs FOR$_INFFORLOP (60=''INFINITE FORMAT LOOP'')
;   333          0395  1 !        SIGNAL_STOPs FOR$_VFEVALERR (68=''VFE VALUE ERROR'')
;   334          0396  1 !--
```

```
  1 !
  2        BEGIN
  2
  2        EXTERNAL REGISTER
  2            CCB : REF $FOR$CCB_DECL,         ! Pointer to Common Control Block
  2            EL_SIZE,                         ! Element size (1st argument)
  2            DT_SEEN,                         ! Data transmitter seen (2nd argument)
  2            FMT_CODE : BLOCK [1, LONG];      ! Format code (return value)
  2
  2        BUILTIN
  2            TESTBITSC;
  2
  2        MACRO                                ! Field definitions for action table
  2            FI_STOP = 0,6,1,0 %,             ! Stop if DT_SEEN
  2            FI_GETW = 0,5,1,0 %,             ! Get w value for format
  2            FI_GETD = 0,4,1,0 %,             ! Get d value for format
  2            FI_GETE = 0,3,1,0 %,             ! Get e value for format
  2            FI_USER = 0,2,1,0 %,             ! Format code involves user data element
  2            FI_EXIT = 0,1,1,0 %,             ! Exit from format interpreter loop
  2            FI_ACTION = 0,0,1,0 %;           ! Code-specific action required;
  2                                             !  see FI_ACT_2 for action
  2
  2                                             ! MAINTENANCE NOTE: An optimization
  2                                             ! below assumes knowledge of certain
  2                                             ! bit settings in FI_ACT.  See comments.
  2
  2        MACRO
M 2            FI_ALL =                         ! Enumerate all format codes
M 2 !                    W D E U E S
M 2 !                          S X T
M 2 !                          E I O
M 2 !                          R T P
M 2              FI_PACK(0,0,0,0,1,1, 0),       ! ER    = 0,     ! 00    ! Format syntax error
M 2              FI_PACK(0,0,0,0,0,0, 2),       ! LP    = 1,     ! 01    ! ( - Format reversin point
M 2              FI_PACK(0,0,0,0,0,0, 3),       ! NLP   = 2,     ! 02    ! n( - Left paren of repeat group
M 2              FI_PACK(0,0,0,0,0,0, 4),       ! RP    = 3,     ! 03    ! ) - Right paren of repeat group
M 2              FI_PACK(0,0,0,0,1,1, 5),       ! EOF   = 4,     ! 04    ! ) - End of format
M 2              FI_PACK(0,0,0,0,1,0, 1),       ! SLS   = 5,     ! 05    ! / - Record separator
M 2              FI_PACK(0,0,0,0,1,0, 1),       ! DLR   = 6,     ! 06    ! $ - Dollar sign: terminal I/O
M 2              FI_PACK(0,0,0,0,1,1, 1),       ! CLN   = 7,     ! 07    ! : - Colon: terminate if end of list
M 2                          0,                         ! UNUSED  8
M 2              FI_PACK(0,0,0,0,0,0, 7),       ! _S    = 9,     ! 09    ! S - Make + optional
M 2              FI_PACK(0,0,0,0,0,0, 8),       ! SP    = 10,    ! 0A    ! SP - Force optional +
M 2              FI_PACK(0,0,0,0,0,0, 7),       ! SS    = 11     ! 0B    ! SS - Leave out optional +
M 2              FI_PACK(1,0,0,0,0,0, 6),       ! _P    = 12,    ! 0C    ! sP - signed scale factor
M 2              FI_PACK(1,0,0,0,0,0,11),       ! _T    = 13,    ! 0D    ! Tn - Tab Set
M 2              FI_PACK(1,0,0,0,1,0, 1),       ! _X    = 14,    ! 0E    old nX
M 2              FI_PACK(1,0,0,0,1,0, 1),       ! _H    = 15,    ! 0F    nHcccc - Hollerith
M 2              FI_PACK(0,0,0,0,0,0, 9),       ! BN    = 16     ! 10    ! BN = Blanks are nulls
M 2              FI_PACK(0,0,0,0,0,0,10),       ! BZ    = 17     ! 11    ! BZ = Blanks are zeroes
M 2              FI_PACK(1,0,0,0,0,0,12),       ! TL    = 18     ! 12    ! TLn = Tab left n columns
M 2              FI_PACK(1,0,0,0,0,0,13),       ! TR    = 19     ! 13    ! TRn (new nX) = Tab right n columns
M 2              FI_PACK(0,0,0,1,1,1, 1),       ! _Q    = 20,    ! 14    ! Q
M 2              FI_PACK(1,0,0,1,1,1, 1),       ! _A    = 21,    ! 15    ! nAw - Alpha numeric
M 2              FI_PACK(1,0,0,1,1,1, 1),       ! _L    = 22,    ! 16    ! nLw - Logical
M 2              FI_PACK(1,0,0,1,1,1, 1),       ! _O    = 23,    ! 17    ! nOw - Octal
M 2              FI_PACK(1,0,0,1,1,1, 1),       ! _I    = 24,    ! 18    ! nIw - Integer
```

```
                FI_PACK(1,0,0,1,1,1, 1),        ! _Z    = 25,   ! 19    ! nZw - Hexadecimal
                FI_PACK(1,1,0,1,1,1, 1),        ! %O    = 26,   ! 1A    ! Ow.m Octal
                FI_PACK(1,1,0,1,1,1, 1),        ! XI    = 27,   ! 1B    ! Iw.m Integer
                FI_PACK(1,1,0,1,1,1, 1),        ! XZ    = 28    ! 1C    ! Zw.m Hexadecimal
                            0,                          ! UNUSED  29
                FI_PACK(1,1,0,1,1,1, 1),        ! _F    = 30,   ! 1E    ! nFw.d - Fixed format
                FI_PACK(1,1,0,1,1,1, 1),        ! _E    = 31,   ! 1F    ! nEw.d - Scientific notation format
                FI_PACK(1,1,0,1,1,1, 1),        ! _G    = 32,   ! 20    ! nGw.d - General format
                FI_PACK(1,1,0,1,1,1, 1),        ! _D    = 33,   ! 21    ! nDw.d - Double Precision format
                FI_PACK(1,1,1,1,1,1, 1),        ! %E    = 34,   ! 22    ! nEw.dEe
                FI_PACK(1,1,1,1,1,1, 1),        ! XG    = 35,   ! 23    ! nGw.dEe
                           0,0,0,0,0,                   ! UNUSED  36:40
                FI_PACK(0,0,0,1,1,1, 1),        ! _DA   = 41,   ! 29    ! nA - default A
                FI_PACK(0,0,0,1,1,1, 1),        ! _DL   = 42,   ! 2A    ! nL - default L
                FI_PACK(0,0,0,1,1,1, 1),        ! _DO   = 43,   ! 2B    ! nO - default O
                FI_PACK(0,0,0,1,1,1, 1),        ! _DI   = 44,   ! 2C    ! nI - default I
                FI_PACK(0,0,0,1,1,1, 1),        ! _DZ   = 45,   ! 2D    ! nZ - default Z
                           0,0,0,0,                     ! UNUSED  46:49
                FI_PACK(0,0,0,1,1,1, 1),        ! _DF   = 50,   ! 32    ! nF - default F
                FI_PACK(0,0,0,1,1,1, 1),        ! _DE   = 51,   ! 33    ! nE - default E
                FI_PACK(0,0,0,1,1,1, 1),        ! _DG   = 52,   ! 34    ! nG - default G
                FI_PACK(0,0,0,1,1,1, 1)         ! _DD   = 53,   ! 35    ! nD - default D
        %;                                      ! End of macro FI_ALL

!+
! Define FI_PACK for use in constructing FI_ACT
!-

        MACRO                                   ! Attributes-packing macro for attributes table
            FI_PACK (W, D, E, U, X, S, NDX) =
                (S^6 + W^5 + D^4 + E^3 + U^2 + X^1 +
                    %IF %IDENTICAL (NDX, 1) %THEN 0 %ELSE 1 %FI) %;

        BIND
            FI_ACT =                            ! First action table
                UPLIT BYTE ( FI_ALL ) : VECTOR [54, BYTE];

!+
! Redefine FI_PACK for use in constructing FI_ACT_2
!-

        UNDECLARE %QUOTE FI_PACK;

        MACRO
            FI_PACK (W, D, E, U, X, S, NDX) =
                NDX %;

        BIND
            FI_ACT_2 =                          ! Second action table
                UPLIT BYTE ( FI_ALL ) : VECTOR [54, BYTE];

!<BLF/PAGE>
```

```
        !+
        ! (NXTITM+1)
        ! Assume that a format code is being repeated.- nf not n(f).
        ! (as distinguished from a repeat group which is n(...))
        ! Decrement format repeat count (ISB$W_FMT_REP). Test
        ! if still more to repeat - if yes, skip usual format code
        ! dispatching by skipping loop altogether, redo defaults if
        ! default format codes and RETURN
        !-

        IF .CCB [ISB$W_FMT_REP] GTR 1
        THEN
            BEGIN

            LOCAL
                ACT : BLOCK [1, LONG];      ! Action table entry for format code

            FMT_CODE = .CCB [ISB$B_FMT_CODE];
            ACT = .FI_ACT [.FMT_CODE];
            IF .DT_SEEN AND .ACT [FI_STOP]
            THEN
                BEGIN
                FMT_CODE = 0;
                RETURN;
                END;
            CCB [ISB$W_FMT_REP] = .CCB [ISB$W_FMT_REP] - 1;
            END
        ELSE

        !+
        ! (FINTRP)
        ! Not in format code repeat - start format interpret loop
        ! Loop until encounter a format code which needs to access
        ! data (ER or explicit or default Q, A, L, O, I, Z, F, E, G, or D),
        !  needs to access the data buffer (X, H, Q), or
        ! depends on whether read or write (), /, $, :, T).
        !-

            BEGIN

            REGISTER
                P,                          ! Pointer to format byte stream
                ACT : BLOCK [1, LONG];      ! Action table entry for format code

            P = .CCB [ISB$A_FMT_PTR];

            DO
                BEGIN

                !+
                ! Pickup next format code byte from compiled format:
                ! If optional representation byte
                ! is present (V_FMT_REPRE=1), mask out flag bit
                ! in format code and copy next byte to BITVECTOR
                ! to indicate larger (less frequent) sizes of the
                ! code representation or Variable Field Expressions (VFE).
```

```
4                 !-
4
4                 FMT_CODE = CH$RCHAR (.P);
4                 FMT_CODE [V_FMT_REPRE] = 0;         ! Clear bit for search
4                 ACT = .FI_ACT [.FMT_CODE];
4
4                 !+
4                 ! If DT_SEEN is set and this format code needs a data transmitter
4                 ! then return a format code of
4                 ! zero to signal the fact.  This will be differentiated from
4                 ! an error by the UDF level by checking DT_SEEN.
4                 !-
4
4                 IF .DT_SEEN AND .ACT [FI_STOP]
4                 THEN
5                     BEGIN
5                     CCB [ISB$A_FMT_PTR] = .P;
5                     FMT_CODE = 0;
5                     RETURN;
4                     END;
4
4                 FMT_CODE = CH$RCHAR_A (P);          ! Re-read and increment pointer
4
4                 !+
4                 ! Optimization:
4                 !
4                 ! Check for certain easily-handled (and frequent) cases:
4                 ! 1.  A/L/O/I/Z (codes 21-25) with no RC and byte-length W;
4                 ! 2.  O/I/Z/F/E/D/G (codes 26-28 and 30-33) with no RC and
4                 !     byte-length W, D;
4                 ! 3.  E/G (codes 34-35) with no RC and byte-length W, D, E;
4                 ! If found, handle directly and bypass the tests for VFE's,
4                 ! word-length RC and W, and special action.  Note that
4                 ! anything with V_FMT_REPRE set falls under OUTRANGE.
4                 !
4                 ! This optimization assumes knowledge of flag bit settings
4                 ! in FI_ACT, and must be reconsidered if FI_ACT changes.
4                 !-
4
5                 IF NOT (CASE .FMT_CODE FROM _A TO XG OF
5                     SET
5                         [_A TO _Z] :
6                             BEGIN
6                             CCB [ISB$W_FMT_W] = RBYTE_A (P);
6                             CCB [ISB$W_FMT_REP] = 1;
6                             CCB [ISB$V_USER_ELEM] = 1;
6                             1               ! Indicate special case found
5                             END;
5                         [XO TO XZ, _F TO _D] :
6                             BEGIN
6                             CCB [ISB$W_FMT_W] = RBYTE_A (P);
6                             CCB [ISB$B_FMT_D] = RBYTE_A (P);
6                             CCB [ISB$B_FMT_E] = 2;
6                             CCB [ISB$W_FMT_REP] = 1;
6                             CCB [ISB$V_USER_ELEM] = 1;
6                             1               ! Indicate special case found
5                             END;
```

```
            [XE TO XG] :
                BEGIN
                CCB [ISB$W_FMT_W] = RBYTE_A (P);
                CCB [ISB$B_FMT_D] = RBYTE_A (P);
                CCB [ISB$B_FMT_E] = RBYTE_A (P);
                CCB [ISB$W_FMT_REP] = 1;
                CCB [ISB$V_USER_ELEM] = 1;
                1               ! Indicate special case found
                END;
            [29, OUTRANGE] :
                0;              ! Indicate special case not found
        TES)
    THEN
        BEGIN

        !+
        ! Get RC, W, D and E in the traditional, fully general
        ! way, including check for VFE's and alternate forms of
        ! W and RC.
        !
        ! Optimization:
        !
        ! If optional second byte is not present, bypass check
        ! for VFE's and alternate forms of W and RC.
        !-

        IF NOT TESTBITSC (FMT_CODE [V_FMT_REPRE])
        THEN
            BEGIN                       ! Begin short form
            CCB [ISB$W_FMT_REP] = 1;
            IF .ACT [FI_GETW]
            THEN
                BEGIN
                CCB [ISB$W_FMT_W] = RBYTE_A (P);
                IF .ACT [FI_GETD]
                THEN
                    BEGIN
                    CCB [ISB$B_FMT_D] = RBYTE_A (P);
                    IF .ACT [FI_GETE]
                    THEN
                        CCB [ISB$B_FMT_E] = RBYTE_A (P);
                    END;
                END;
            END                         ! End short form
        ELSE
            BEGIN                       ! Begin long form

            LOCAL
                FMT_REPR : BLOCK [1, LONG];

            FMT_REPR = RBYTE_A (P);

            !+
            ! Get repeat count (RC) from format and save in ISB$W_FMT_REP.
            ! If repeat count is a VFE (FMT_REPR[V_RC_VFE]=1), get VFE and
            ! check for out of range (1:32767).
            ! If explicitly represented, get byte or word value.
```

```
6          ! Else set repeat count to 1. Possible for left paren
6          ! of a repeat group (NLP) or A, L, O, Z, I, F, E, G, D
6          ! or default A, L, O, Z, I, F, E, G, D.
6          !-
6
7          CCB [ISB$W_FMT_REP] = (IF .FMT_REPR [V_RC_VFE]
7              THEN
8                  BEGIN                   ! Process RC VFE
8
8                  LOCAL
8                      T;
8
8                  T = CALL_VFE (P);
8
8                  IF .T GEQU 32768 OR .T EQL 0
8                  THEN
9                      BEGIN
9                      FOR$$SIGNAL (FOR$K_VFEVALERR);
9                      1                   ! Force repeat count to 1 on error
9                      END
8                  ELSE
8                      .T                  ! Use user supplied value
8                  END                     ! End of RC VFE processing
8
7              ELSE
7
7                  ! The following assumes that RC is either 00
7                  ! (absent), 01 (byte) or 10 (word), and that
7                  ! it cannot be 11.
7
7                  IF .FMT_REPR [V_RC_TYPE_BYTE]
7                  THEN
8                      RBYTE_A (P)                 ! RC is a byte
7                  ELSE
7                      IF .FMT_REPR [V_RC_TYPE_WORD]
7                      THEN
8                          RWORD_A (P)             ! RC is a word
7                      ELSE
6                          1);                     ! RC is absent
6
6          !+
6          ! P, T, X, H, A ,L, O, I, Z, F, E, G, D:
6          ! Get field width (w) from format and
6          ! set ISB$W_FMT_W. If width field is a
6          ! VFE (V_W_VFE=T), get VFE value and check range;
6          ! if P scale -128 to 127, else (field width w) 0 to 32767.
6          ! If width of field is a byte (V_W_WORD=0), get byte
6          ! else get word.  ISB$W_FMT_W is set as a
6          ! zero extended word.
6          !-
6
6          IF .ACT [FI_GETW]
6          THEN
7              BEGIN
8              CCB [ISB$W_FMT_W] = (IF .FMT_REPR [V_W_VFE] THEN
9                  BEGIN
9
```

```
    LOCAL
        T;

    T = CALL_VFE (P);

    IF .FMT_CODE EQL _P
    THEN
        BEGIN                   ! P scale

        IF .T<0,8,1> NEQ .T     ! P between -128 and 127?
        THEN
            BEGIN
            FOR$$SIGNAL (FOR$K_VFEVALERR);
            T = 0                       ! Force P scale to 0
            END

        END
    ELSE
        BEGIN                           ! Else w width of field

        IF .T GEQU 32768
        THEN
            BEGIN
            FOR$$SIGNAL (FOR$K_VFEVALERR);
            T = 1
            END

        END;

    .T                          ! return VFE value
    END
ELSE IF .FMT_REPR [V_W_WORD] THEN RWORD_A (P) ELSE RBYTE_A (P));

!+
! Get decimal part (d) from format and set
! ISB$B_FMT_D. If decimal part is a VFE
! (V_D_VFE=T) get VFE and check range (0:32767).
! Else get byte from format
!
! Set default exponent width to 2.
!-

IF .ACT [FI_GETD]
THEN
    BEGIN
    CCB [ISB$B_FMT_D] = (IF .FMT_REPR [V_D_VFE] THEN
        BEGIN                   ! VFE

        LOCAL
            T;

        T = CALL_VFE (P);

        IF .T GEQU 32768
        THEN
            BEGIN
            FOR$$SIGNAL (FOR$K_VFEVALERR);
```

```
                                                        1
                                                        END
                                                ELSE
                                                        .T

                                                END
                                        ELSE RBYTE_A (P));
                                        CCB [ISB$B_FMT_E] = 2;

                                        !+
                                        ! Get exponent width (e) from format and set
                                        ! ISB$B_FMT_E.  If exponent width is a VFE, check
                                        ! range (0:255).  Else get byte from format.
                                        !-

                                        IF .ACT [FI_GETE]
                                        THEN
                                            BEGIN
                                            CCB [ISB$B_FMT_E] = (IF .FMT_REPR [V_E_VFE] THEN
                                                BEGIN                  ! VFE

                                                LOCAL
                                                    T;

                                                T = CALL_VFE (P);

                                                IF .T GEQU 256
                                                THEN
                                                    BEGIN
                                                    FOR$$SIGNAL (FOR$K_VFEVALERR);
                                                    1
                                                    END
                                                ELSE
                                                    .T

                                                END
                                            ELSE RBYTE_A (P));
                                            END;

                                        END;

                                    END;

                                END;                                ! End long form

                            !+
                            ! For all user data element format codes
                            ! (explicit and default Q, A, L, O, I, Z, F, E, G, D):
                            ! Set user data element format code
                            ! seen in this group, because not in an
                            ! infinite format loop invoking for a user
                            ! data element format code which doesn't exist.
                            !-

                            IF .ACT [FI_USER] THEN CCB [ISB$V_USER_ELEM] = 1;

                            !+
                        ! Dispatch on format code and select appropriate actions:
                        !-

                        IF .ACT [FI_ACTION]
                        THEN
                            CASE .FI_ACT_2 [.FMT_CODE] FROM 0 TO 13 OF
                            SET

                            [0] :

                                !+
                                ! ER or undefined format code
                                ! Bad format: Signal_stop SYNTAX ERROR IN FORMAT (FOR$_SYNERRFOR)
                                !-

                                BEGIN
                                FOR$$SIGNAL_STO (FOR$K_SYNERRFOR);
                                FMT_CODE = 0;
                                RETURN;
                                END;

                            [1] :

                                !+
                                ! No special actions required.
                                !-

                                ;

                            [2] :

                                !+
                                ! LP  Format reversion point: left paren of
                                ! second outer-most pair. Remeber current format
                                ! offset (ISB$W_FMT_REVER) in case more data
                                ! element in I/O list than data format
                                ! codes in format. Reset push down stack to
                                ! empty (-1) since this is start of
                                ! first group repeat. Clear user data element
                                ! seen flag (ISB$V_USER_ELEM) as a defense
                                ! against infinite loop with no data
                                ! transmit format code
                                ! Note: format text pointer already advanced to next byte
                                !-

                                BEGIN
                                CCB [ISB$B_FMT_DEP] = -1;
                                CCB [ISB$W_FMT_REVER] = .P - .CCB [ISB$A_FMT_BEG];
                                CCB [ISB$V_USER_ELEM] = 0;
                                END;                        ! End LP

                            [3] :

                                !+
                                ! NLP   Left paren of a repeat group: Push repeat
                                ! count (ISB$W_FMT_REP) and current (ISB$A_FMT_PTR)
                                ! onto format stacks
```

```
5                !-
6
6                BEGIN
6                CCB [ISB$B_FMT_DEP] = .CCB [ISB$B_FMT_DEP] + 1;
6                VECTOR [CCB [ISB$W_FMT_STKR], .CCB [ISB$B_FMT_DEP];, WORD, UNSIGNED]    !
6                = .CCB [ISB$W_FMT_REP];
6                VECTOR [CCB [ISB$W_FMT_STKP], .CCB [ISB$B_FMT_DEP];, WORD, UNSIGNED]    !
6                = .P - .CCB [ISB$A_FMT_BEG];
6                CCB [ISB$W_FMT_REP] = T;
5                END;                    ! End NLP
5
5            [4] :
5
5                !+
5                ! RP   Right paren of repeat group: Decrement
5                ! current group repeat count (on top of
5                ! stack)  If current group count still greater
5                ! than 0, set current format pointer back to
5                ! beginning of repeat group. Else pop off
5                ! beginning of group pointer and group repeat count
5                !-
5
6                IF (VECTOR [CCB [ISB$W_FMT_STKR], .CCB [ISB$B_FMT_DEP];, WORD, UNSIGNED]        !
5                    = .VECTOR [CCB [ISB$W_FMT_STKR], .CCB [ISB$B_FMT_DEP];, WORD, UNSIGNED] - 1) GTR
5                THEN
5                ! reset pointer to address of repeat group
5                !
5                    P = .CCB [ISB$A_FMT_BEG]     !
5                    + .VECTOR [CCB [ISB$W_FMT_STKP], .CCB [ISB$B_FMT_DEP];, WORD, UNSIGNED]
5                ELSE
5                ! pop off pointer and repeat count
5                !
5                    CCB [ISB$B_FMT_DEP] = .CCB [ISB$B_FMT_DEP] - 1;
5
5            [5] :
5
5                !+
5                ! EOF   End of format:
5                ! If not end of user I/O list (EL_SIZE=0)
5                ! and no user data element format code
5                ! (ISB$V_USER_ELEM=0), then Signal_stop. INFINITE
5                ! FORMAT LOOP (FOR$_INFFORLOP).
5                ! Reset current format pointer to reversion point
5                ! (ISB$W_FMT_REVER). Initialize format stack depth.
5                !-
5
6                BEGIN
6                P = .CCB [ISB$A_FMT_BEG] + .CCB [ISB$W_FMT_REVER];
6                CCB [ISB$B_FMT_DEP] = -1;
6
6                IF .EL_SIZE GTRU 0 AND NOT .CCB [ISB$V_USER_ELEM]
6                THEN
7                    BEGIN
7                    FOR$$SIGNAL_STO (FOR$K_INFFORLOO);
7                    FMT_CODE = 0;
7                    RETURN;
6                    END;
```

```
    END;
[6] :

    !+
    ! P   Scale factor (sP): -128 =< s =< 127
    ! Convert unsigned word width (w) (ISB$W_FMT_W)
    ! to signed byte ('s) and save in ISB$B_FMT_P.
    !-

    BEGIN
    CCB [ISB$B_FMT_P] = .CCB [ISB$W_FMT_W];
    END;
[7] :

    !+
    ! S, SS   Restore option of + to processor.
    !-

    BEGIN
    CCB [ISB$V_SP] = 0;
    END;
[8] :

    !+
    ! SP   Force optional + to appear
    !-

    BEGIN
    CCB [ISB$V_SP] = 1;
    END;
[9] :

    !+
    ! BN   Treat blanks as nulls on numeric input.
    !-

    BEGIN
    CCB [ISB$V_BN] = 1;
    END;
[10] :

    !+
    ! BZ   Treat blanks as zeroes on numeric input.
    !-

    BEGIN
    CCB [ISB$V_BN] = 0;
    END;


[11] :
```

```
5
5                   !+
5                   ! Tn   Move buffer pointer to position n
5                   !-
5
6                   BEGIN
6                   CCB [LUB$A_BUF_PTR] = .CCB [LUB$A_BUF_BEG] + (.CCB [ISB$W_FMT_W] - 1);
5                   END;
5
5               [12] :
5
5                   !+
5                   ! TLn  Move buffer pointer left n positions
5                   !-
5
6                   BEGIN
6                   CCB [LUB$A_BUF_PTR] = .CCB [LUB$A_BUF_PTR] - .CCB [ISB$W_FMT_W];
6
6                   IF .CCB [LUB$A_BUF_PTR] LSSA .CCB [LUB$A_BUF_BEG]
6                   THEN
6                       CCB [LUB$A_BUF_PTR] = .CCB [LUB$A_BUF_BEG];
6
5                   END;
5
5               [13] :
5
5                   !+
5                   ! TRn  Move buffer pointer right n spaces.
5                   !           Note: as of VMS Release 2, the format nX
5                   !           is equivalent to TRn.  The old nX code
5                   !           is no longer generated but is supported
5                   !           for compatibility.
5                   !-
5
6                   BEGIN
6                   CCB [LUB$A_BUF_PTR] = .CCB [LUB$A_BUF_PTR] + .CCB [ISB$W_FMT_W];
5                   END;
5               TES;
5
5           !+
5           ! End of loop - continue if just format control
5           ! ( (, n(, )) or not dependent on read/write
5           ! and doesn't access data buffer (P)
5           !
5           ! EXITLOOP for format codes which access user data.
5           ! (ER or explicit or  default A, L, O, I, Z, F, E, G or D),
5           ! EXITLOOP for format codes which access data
5           ! buffer (X, H, Q)  EXITLOOP for format codes
5           ! which depend on whether read or write (end
5           ! of format, /, $, :, T).
5           !-
5
5           END
4       END
3   UNTIL .ACT [FI_EXIT];
3
3   !+
3          ! Reset format code and updated format text pointer in ISB.
3          !-
3
3          CCB [ISB$B_FMT_CODE] = .FMT_CODE;
3          CCB [ISB$A_FMT_PTR] = .P;
2          END;
2
```

```
	!+
	! Default data format codes - set defaults based on size of
	! each user data element even if inside a format code repeat
	! since the size could be different for each user data element
	!-

	IF .FMT_CODE GEQU _DA
	THEN
	    BEGIN

	    CASE .FMT_CODE FROM _DA TO _DD OF
		SET

		[_DA] :

		    !+
		    ! Default A: set w field (ISB$W_FMT_W) from
		    ! size of user data element
		    !-

		    CCB [ISB$W_FMT_W] = .EL_SIZE;

		[_DL] :

		    !+
		    ! Default L:  set w field (ISB$W_FMT_W) to 2
		    !-

		    CCB [ISB$W_FMT_W] = 2;

		[_DI] :

		    !+
		    ! Default I: Set w field to 7 if element is smaller than
		    ! 4 bytes else set it to 12.
		    !-

		    IF .EL_SIZE LSSU 4 THEN CCB [ISB$W_FMT_W] = 7 ELSE CCB [ISB$W_FMT_W] = 12;

		[_DO, _DZ] :

		    !+
		    ! Default O, Z.  Set to the width that would allow O
		    ! format plus a space.  \\ Note:  For compatibility with
		    ! previous releases, the sizes for 1, 2 and 4 bytes must
		    ! be 7, 7 and 12 respectively. \\
		    !-

		    CCB [ISB$W_FMT_W] = MAX (7, MIN (65535, (((8*.EL_SIZE)+2)/3)+1));
		[_DF, _DE, _DG, _DD] :

		    !+
		    ! Default F, E, G, D: Set w and e fields as is appropriate
		    ! to the element size.  Note that anything that is not
		    ! 8 (REAL*8) or 16 (REAL*16) is assumed to be 4 (REAL*4),
		    ! but check for 4 first.
		    !-
```

```
                BEGIN

                SELECTONE .EL_SIZE OF
                    SET

                    [4] :
                        BEGIN
                        CCB [ISB$B_FMT_E] = 2;
                        CCB [ISB$W_FMT_W] = 15;
                        CCB [ISB$B_FMT_D] = 7;
                        END;

                    [8] :
                        BEGIN
                        CCB [ISB$B_FMT_E] = 2;
                        CCB [ISB$W_FMT_W] = 25;
                        CCB [ISB$B_FMT_D] = 16;
                        END;

                    [16] :
                        BEGIN
                        CCB [ISB$B_FMT_E] = 3;
                        CCB [ISB$W_FMT_W] = 42;
                        CCB [ISB$B_FMT_D] = 33;
                        END;

                    [OTHERWISE] :
                        BEGIN
                        CCB [ISB$B_FMT_E] = 2;
                        CCB [ISB$W_FMT_W] = 15;
                        CCB [ISB$B_FMT_D] = 7;
                        END;
                    TES;

                END;

            [INRANGE] :
                ;
            TES;

        !+
        ! Translate default format code to corresponding explicit code.
        !-

        FMT_CODE = .FMT_CODE - (_DA - _A);
        END;

    !+
    ! Return to read, write User Data Formatter (UDF). If default
    ! format code, return corresponding explicit format code
    ! to UDF. Else return the actual format code
    !-

    RETURN;

    END;                                ! End of routine FOR$$FMT_INTRP1
```

```
FO
Sy

FO
SY

PS
--
.
_F

Ph
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As

Th
16
Th
13
2

Ma
--
_$

13

Th

MA
```

```
22 21 21 01 01 01 00 42 02 02 43 01 01 01 43 00026 P.AAA:  .BYTE   67, 1, 1, 1, 67, 2, 2, 66, 0, 1, 1, 1, -           ;
00 76 76 76 66 66 66 66 66 46 21 21 01 01 22 00035                 33, 33, 34, 34, 1, 1, 33, 33, 70, 102, -          ;
46 46 46 46 00 00 00 00 00 7E 7E 76 76 76 76 00044                 102, 102, 102, 102, 118, 118, 118, 0, -           ;
                  46 46 46 46 00 00 00 00 46 00053                 118, 118, 118, 118, 126, 126, 0, 0, 0, 0, -       ;
                                                                   0, 70, 70, 70, 70, 70, 0, 0, 0, 0, 70, -          ;
                                                                   70, 70, 70                                        ;
01 0B 06 07 08 07 00 01 01 01 05 04 03 02 00 0005C P.AAB:  .BYTE   0, 2, 3, 4, 5, 1, 1, 1, 0, 7, 8, 7, 6, -          ;
00 01 01 01 01 01 01 01 01 01 0D 0C 0A 09 01 0006B                 11, 1, 1, 9, 10, 12, 13, 1, 1, 1, 1, 1, -         ;
01 01 01 01 00 00 00 00 00 01 01 01 01 01 01 0007A                 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, -       ;
                  01 01 01 01 00 00 00 00 01 00089                 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, -       ;
                                                                   1                                                 ;

                                                   FI_ACT=                 P.AAA
                                                   FI_ACT_2=               P.AAB

                         5E              08  C2 00000 FOR$$FMT_INTRP1::
                                                                 SUBL2   #8, SP                                      ; 0274
                                    8D   AB  9F 00003            PUSHAB  -115(CCB)                                   ; 0517
                         01         00   BE  B1 00006            CMPW    @0(SP), #1                                  ;
                                         19  15 0000A            BLEQ    2$                                          ;
                         58         8F   AB  9A 0000C            MOVZBL  -113(CCB), FMT_CODE                         ; 0524
                         50         80 AF48  9A 00010            MOVZBL  FI_ACT[FMT_CODE], ACT                       ; 0525
                         07              59  E9 00015            BLBC    DT_SEEN, 1$                                 ; 0526
                03       50              06  E1 00018            BBC     #6, ACT, 1$                                 ;
                                       0258  31 0001C            BRW     41$                                         ;
                                    00   BE  B7 0001F 1$:        DECW    @0(SP)                                      ; 0532
                                       02B2  31 00022            BRW     52$                                         ; 0517
                         52         80   AB  D0 00025 2$:        MOVL    -128(CCB), P                                ; 0551
                         58              62  9A 00029 3$:        MOVZBL  (P), FMT_CODE                               ; 0565
                         58         80   8F  8A 0002C            BICB2   #128, FMT_CODE                              ; 0566
                         53       FF5F CF48  9A 00030            MOVZBL  FI_ACT[FMT_CODE], ACT                       ; 0567
                         0B              59  E9 00036            BLBC    DT_SEEN, 4$                                 ; 0576
                07       53              06  E1 00039            BBC     #6, ACT, 4$                                 ;
                    80   AB              52  D0 0003D            MOVL    P, -128(CCB)                                ; 0579
                                       0233  31 00041            BRW     41$                                         ; 0580
                         58              82  9A 00044 4$:        MOVZBL  (P)+, FMT_CODE                              ; 0584
                0E       15              58  CF 00047            CASEL   FMT_CODE, #21, #14                          ; 0602
      0020    0020     0020            0020     0004B 5$:        .WORD   6$-5$,-                                     ;
      0026    0026     0026            0020     00053                    6$-5$,-                                     ;
      0026    0026     0026            0047     0005B                    6$-5$,-                                     ;
              0034     0034            0026     00063                    6$-5$,-                                     ;
                                                                         6$-5$,-                                     ;
                                                                         7$-5$,-                                     ;
                                                                         7$-5$,-                                     ;
                                                                         7$-5$,-                                     ;
                                                                         10$-5$,-                                    ;
                                                                         7$-5$,-                                     ;
                                                                         7$-5$,-                                     ;
                                                                         7$-5$,-                                     ;
                                                                         7$-5$,-                                     ;
                                                                         8$-5$,-                                     ;
                                                                         8$-5$                                       ;
                                         27  11 00069            BRB     10$                                         ;
```

```
                          89    AB          82  9B 0006B 6$:     MOVZBW  (P)+, -119(CCB)                            ; 0606
                                            16  11 0006F         BRB     9$                                         ; 0607
                          89    AB          82  9B 00071 7$:     MOVZBW  (P)+, -119(CCB)                            ; 0613
                          8B    AB          82  90 00075         MOVB    (P)+, -117(CCB)                            ; 0614
                          8C    AB          02  90 00079         MOVB    #2, -116(CCB)                              ; 0615
                                            08  11 0007D         BRB     9$                                         ; 0616
                          89    AB          82  9B 0007F 8$:     MOVZBW  (P)+, -119(CCB)                            ; 0622
                          8B    AB          82  B0 00083         MOVW    (P)+, -117(CCB)                            ; 0623
                          00    BE          01  B0 00087 9$:     MOVW    #1, @0(SP)                                 ; 0625
                          96    AB          08  88 0008B         BISB2   #8, -106(CCB)                              ; 0626
                                          0236  31 0008F         BRW     50$                                        ;
               28               58          07  E4 00092 10$:    BBSC    #7, FMT_CODE, 14$                          ; 0646
                          00    BE          01  B0 00096         MOVW    #1, @0(SP)                                 ; 0649
               03               53          05  E0 0009A         BBS     #5, ACT, 11$                               ; 0650
                                          012A  31 0009E         BRW     30$                                        ;
                          89    AB          82  9B 000A1 11$:    MOVZBW  (P)+, -119(CCB)                            ; 0653
               03               53          04  E0 000A5         BBS     #4, ACT, 12$                               ; 0654
                                          011F  31 000A9         BRW     30$                                        ;
                          8B    AB          82  90 000AC 12$:    MOVB    (P)+, -117(CCB)                            ; 0657
               03               53          03  E0 000B0         BBS     #3, ACT, 13$                               ; 0658
                                          0114  31 000B4         BRW     30$                                        ;
                          8C    AB          82  90 000B7 13$:    MOVB    (P)+, -116(CCB)                            ; 0660
                                          010D  31 000BB         BRW     30$                                        ; 0646
                          08    AE          82  9A 000BE 14$:    MOVZBL  (P)+, FMT_REPR                             ; 0670
                                      08    AE  95 000C2         TSTB    FMT_REPR                                   ; 0682
                                            21  18 000C5         BGEQ    16$                                        ;
                                50          82  D0 000C7         MOVL    (P)+, T                                    ; 0689
                              6240          00  FB 000CA         CALLS   #0, (P)[T]                                 ;
                   00008000     8F          50  D1 000CE         CMPL    T, #32768                                  ; 0691
                                            04  1E 000D5         BGEQU   15$                                        ;
                                            50  D5 000D7         TSTL    T                                          ;
                                            23  12 000D9         BNEQ    19$                                        ;
                                7E    44    8F  9A 000DB 15$:    MOVZBL  #68, -(SP)                                 ; 0694
                   00000000G    00          01  FB 000DF         CALLS   #1, FOR$$SIGNAL                            ;
                                            13  11 000E6         BRB     18$                                        ; 0693
                                05    08    AE  E9 000E8 16$:    BLBC    FMT_REPR, 17$                              ; 0707
                                50          82  9A 000EC         MOVZBL  (P)+, R0                                   ; 0709
                                            0D  11 000EF         BRB     19$                                        ;
               05         08    AE          01  E1 000F1 17$:    BBC     #1, FMT_REPR, 18$                          ; 0711
                                50          82  3C 000F6         MOVZWL  (P)+, R0                                   ; 0713
                                            03  11 000F9         BRB     19$                                        ;
                                50          01  D0 000FB 18$:    MOVL    #1, R0                                     ; 0711
                          00    BE          50  B0 000FE 19$:    MOVW    R0, @0(SP)                                 ; 0682
               03               53          05  E0 00102         BBS     #5, ACT, 20$                               ; 0728
                                          00C2  31 00106         BRW     30$                                        ;
               48         08    AE          06  E1 00109 20$:    BBC     #6, FMT_REPR, 23$                          ; 0731
                                50          82  D0 0010E         MOVL    (P)+, T                                    ; 0737
                              6240          00  FB 00111         CALLS   #0, (P)[T]                                 ;
                          04    AE          50  D0 00115         MOVL    R0, T                                      ;
                                0C          58  D1 00119         CMPL    FMT_CODE, #12                              ; 0739
                                            19  12 0011C         BNEQ    21$                                        ;
    04    AE      04    AE      08          00  EC 0011E         CMPV    #0, #8, T, T                               ; 0743
                                            29  13 00125         BEQL    22$                                        ;
                                7E    44    8F  9A 00127         MOVZBL  #68, -(SP)                                 ; 0746
                   00000000G    00          01  FB 0012B         CALLS   #1, FOR$$SIGNAL                            ;
                                      04    AE  D4 00132         CLRL    T                                          ; 0747
                                            19  11 00135         BRB     22$                                        ; 0741
```

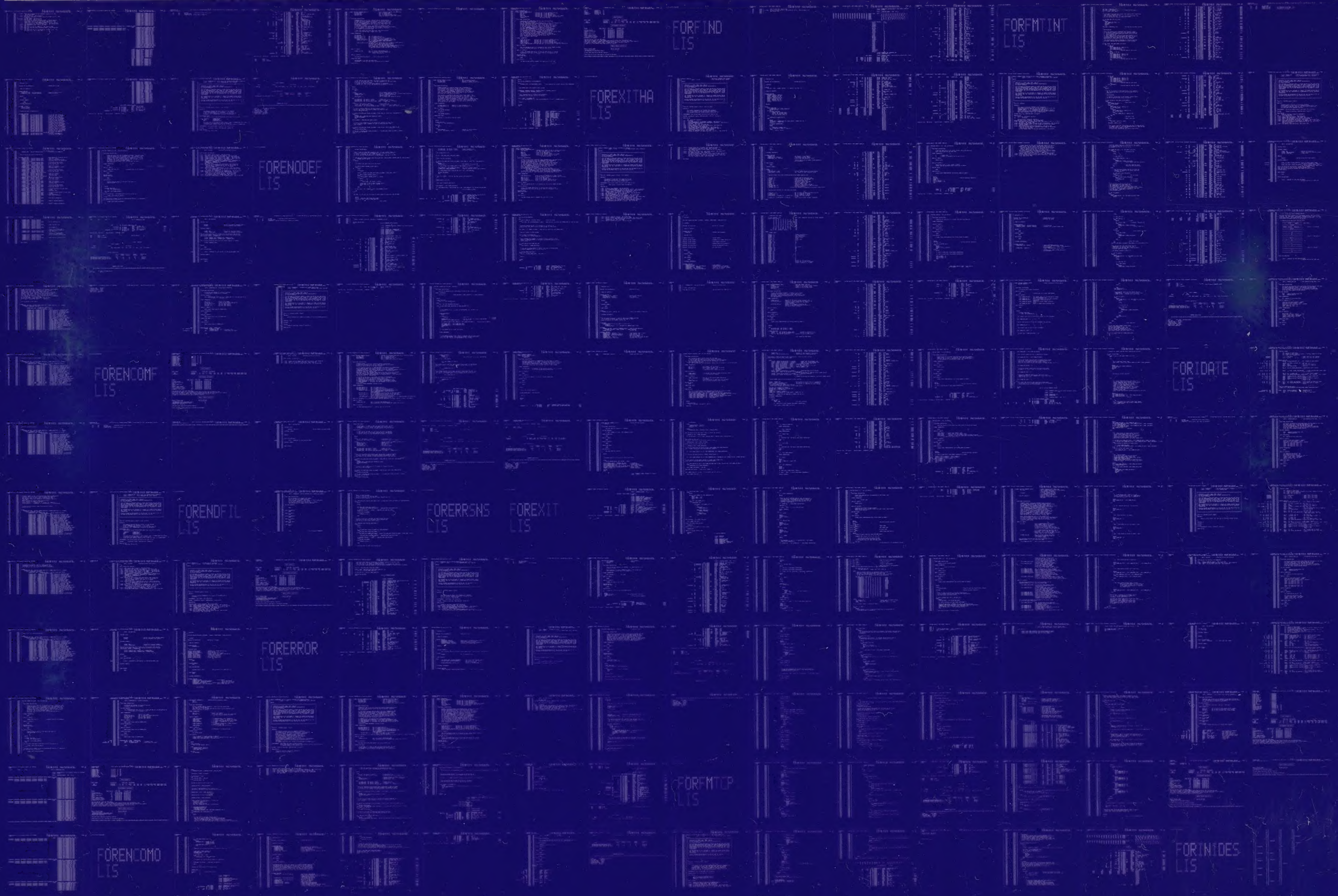
```
                    00008000   8F      04  AE  D1 00137 21$:   CMPL    T, #32768                      ; 0754
                                           0F  1F 0013F        BLSSU   22$                            ;
                               7E      44  8F  9A 00141        MOVZBL  #68, -(SP)                     ; 0757
                   00000000G   00          01  FB 00145        CALLS   #1, FOR$$SIGNAL                ;
                          04   AE          01  D0 0014C        MOVL    #1, T                          ; 0758
                               50      04  AE  D0 00150 22$:   MOVL    T, R0                          ; 0763
                                           0D  11 00154        BRB     25$                            ;
             05           08   AE          02  E1 00156 23$:   BBC     #2, FMT_REPR, 24$              ; 0765
                               50          82  3C 0015B        MOVZWL  (P)+, R0                       ;
                                           03  11 0015E        BRB     25$                            ;
                               50          82  9A 00160 24$:   MOVZBL  (P)+, R0                       ;
                          89   AB          50  B0 00163 25$:   MOVW    R0, -119(CCB)                  ; 0731
             60                53          04  E1 00167        BBC     #4, ACT, 30$                   ; 0776
             20           08   AE          05  E1 0016B        BBC     #5, FMT_REPR, 26$              ; 0779
                               50          82  D0 00170        MOVL    (P)+, T                        ; 0785
                             6240          00  FB 00173        CALLS   #0, (P)[T]                     ;
                    00008000   8F          50  D1 00177        CMPL    T, #32768                      ; 0787
                                           13  1F 0017E        BLSSU   27$                            ;
                               7E      44  8F  9A 00180        MOVZBL  #68, -(SP)                     ; 0790
                   00000000G   00          01  FB 00184        CALLS   #1, FOR$$SIGNAL                ;
                               50          01  D0 0018B        MOVL    #1, R0                         ; 0789
                                           03  11 0018E        BRB     27$                            ; 0787
                               50          82  9A 00190 26$:   MOVZBL  (P)+, R0                       ; 0797
                          8B   AB          50  90 00193 27$:   MOVB    R0, -117(CCB)                  ; 0779
                          8C   AB          02  90 00197        MOVB    #2, -116(CCB)                  ; 0798
             2C                53          03  E1 0019B        BBC     #3, ACT, 30$                   ; 0806
             20           08   AE          04  E1 0019F        BBC     #4, FMT_REPR, 28$              ; 0809
                               50          82  D0 001A4        MOVL    (P)+, T                        ; 0815
                             6240          00  FB 001A7        CALLS   #0, (P)[T]                     ;
                    00000100   8F          50  D1 001AB        CMPL    T, #256                        ; 0817
                                           13  1F 001B2        BLSSU   29$                            ;
                               7E      44  8F  9A 001B4        MOVZBL  #68, -(SP)                     ; 0820
                   00000000G   00          01  FB 001B8        CALLS   #1, FOR$$SIGNAL                ;
                               50          01  D0 001BF        MOVL    #1, R0                         ; 0819
                                           03  11 001C2        BRB     29$                            ; 0817
                               50          82  9A 001C4 28$:   MOVZBL  (P)+, R0                       ; 0827
                          8C   AB          50  90 001C7 29$:   MOVB    R0, -116(CCB)                  ; 0809
             04                53          02  E1 001CB 30$:   BBC     #2, ACT, 31$                   ; 0845
                          96   AB          08  88 001CF        BISB2   #8, -106(CCB)                  ;
                               03          53  E8 001D3 31$:   BLBS    ACT, 32$                       ; 0851
                                         00EF  31 001D6        BRW     50$                            ;
             0D                00    FDEC CF48 8F 001D9 32$:   CASEB   FI_ACT_2[FMT_CODE], #0, #13    ; 0853
      0032   0020            00E8         001C    001E0 33$:   .WORD   34$-33$,-                      ;
      00A3   009C            0078         0050    001E8                50$-33$,-                      ;
      00BB   00B5            00AF         00A9    001F0                35$-33$,-                      ;
                             00E0         00CA    001F8                36$-33$,-                      ;
                                                                       37$-33$,-                      ;
                                                                       39$-33$,-                      ;
                                                                       42$-33$,-                      ;
                                                                       43$-33$,-                      ;
                                                                       44$-33$,-                      ;
                                                                       45$-33$,-                      ;
                                                                       46$-33$,-                      ;
                                                                       47$-33$,-                      ;
                                                                       48$-33$,-                      ;
                                                                       49$-33$                        ;
                                            3E  DD 001FC 34$:  PUSHL   #62                            ; 0864
```

```
                                          70   11 001FE         BRB     40$                                   ;
                      92   AB             01   8E 00200 35$:    MNEGB   #1, -110(CCB)                         ; 0894
     90   AB          52         FF7C     CB   A3 00204         SUBW3   -132(CCB), P, -112(CCB)               ; 0895
                      96   AB             08   8A 0020B         BICB2   #8, -106(CCB)                         ; 0896
                                        00B6   31 0020F         BRW     50$                                   ; 0853
                                    92    AB   96 00212 36$:    INCB    -110(CCB)                             ; 0908
                           50       92    AB   9A 00215         MOVZBL  -110(CCB), R0                         ; 0909
                    FF60 CB40       00    BE   B0 00219         MOVW    @0(SP), -160(CCB)[R0]                 ; 0910
     FF50 CB40             52     FF7C    CB   A3 00220         SUBW3   -132(CCB), P, -176(CCB)[R0]           ; 0912
                      00   BE             01   B0 00229         MOVW    #1, @0(SP)                            ; 0913
                                        0098   31 0022D         BRW     50$                                   ; 0853
                           50       92    AB   9A 00230 37$:    MOVZBL  -110(CCB), R0                         ; 0927
                           51     FF60  CB40   3C 00234         MOVZWL  -160(CCB)[R0], R1                     ; 0928
                                          51   D7 0023A         DECL    R1                                    ;
                    FF60 CB40             51   B0 0023C         MOVW    R1, -160(CCB)[R0]                     ;
                                          51   D5 00242         TSTL    R1                                    ;
                                          0D   15 00244         BLEQ    38$                                   ;
                           52     FF50  CB40   3C 00246         MOVZWL  -176(CCB)[R0], P                      ; 0933
                           52     FF7C    CB   C0 0024C         ADDL2   -132(CCB), P                          ;
                                          75   11 00251         BRB     50$                                   ; 0932
                                    92    AB   97 00253 38$:    DECB    -110(CCB)                             ; 0937
                                          70   11 00256         BRB     50$                                   ; 0927
                           52       90    AB   3C 00258 39$:    MOVZWL  -112(CCB), P                          ; 0952
                           52     FF7C    CB   C0 0025C         ADDL2   -132(CCB), P                          ;
                      92   AB             01   8E 00261         MNEGB   #1, -110(CCB)                         ; 0953
                                          5A   D5 00265         TSTL    EL_SIZE                               ; 0955
                                          5F   13 00267         BEQL    50$                                   ;
          5A          96   AB             03   E0 00269         BBS     #3, -106(CCB), 50$                    ;
                                          3C   DD 0026E         PUSHL   #60                                   ; 0958
               00000000G   00             01   FB 00270 40$:    CALLS   #1, FOR$$SIGNAL_STO                   ;
                                          58   D4 00277 41$:    CLRL    FMT_CODE                              ; 0959
                                        00F4   31 00279         BRW     66$                                   ; 0957
                      88   AB       89    AB   90 0027C 42$:    MOVB    -119(CCB), -120(CCB)                  ; 0974
                                          45   11 00281         BRB     50$                                   ; 0853
                      94   AB             01   8A 00283 43$:    BICB2   #1, -108(CCB)                         ; 0984
                                          3F   11 00287         BRB     50$                                   ; 0853
                      94   AB             01   88 00289 44$:    BISB2   #1, -108(CCB)                         ; 0994
                                          39   11 0028D         BRB     50$                                   ; 0853
                      93   AB             01   88 0028F 45$:    BISB2   #1, -109(CCB)                         ; 1004
                                          33   11 00293         BRB     50$                                   ; 0853
                      93   AB             01   8A 00295 46$:    BICB2   #1, -109(CCB)                         ; 1014
                                          2D   11 00299         BRB     50$                                   ; 0853
                           50       89    AB   3C 0029B 47$:    MOVZWL  -119(CCB), R0                         ; 1025
                           50       BC    AB   C0 0029F         ADDL2   -68(CCB), R0                          ;
                      B0   AB       FF    A0   9E 002A3         MOVAB   -1(R0), -80(CCB)                      ;
                                          1E   11 002A8         BRB     50$                                   ; 0853
                           50       89    AB   3C 002AA 48$:    MOVZWL  -119(CCB), R0                         ; 1035
                      B0   AB             50   C2 002AE         SUBL2   R0, -80(CCB)                          ;
                      BC   AB       B0    AB   D1 002B2         CMPL    -80(CCB), -68(CCB)                    ; 1037
                                          0F   1E 002B7         BGEQU   50$                                   ;
                      B0   AB       BC    AB   D0 002B9         MOVL    -68(CCB), -80(CCB)                    ; 1039
                                          08   11 002BE         BRB     50$                                   ; 0853
                           50       89    AB   3C 002C0 49$:    MOVZWL  -119(CCB), R0                         ; 1054
                      B0   AB             50   C0 002C4         ADDL2   R0, -80(CCB)                          ;
          03               53             01   E0 002C8 50$:    BBS     #1, ACT, 51$                          ; 1073
                                        FD5A   31 002CC         BRW     3$                                    ;
                      8F   AB             58   90 002CF 51$:    MOVB    FMT_CODE, -113(CCB)                   ; 1079
```

```
                             80      AB               52  D0 002D3          MOVL    P, -128(CCB)                         ; 1080
                                     29               58  D1 002D7  52$:    CMPL    FMT_CODE, #41                        ; 1089
                                                      03  1E 002DA          BGEQU   53$                                  ;
                                                    0091  31 002DC          BRW     66$                                  ;
                     0C              29               58  CF 002DF  53$:    CASEL   FMT_CODE, #41, #12                   ; 1093
       0026        0037            0020             001A     002E3  54$:    .WORD   55$-54$,-                            ;
       008A        008A            008A             0037     002EB                  56$-54$,-                            ;
       005F        005F            005F             008A     002F3                  59$-54$,-                            ;
                                                    005F     002FB                  57$-54$,-                            ;
                                                                                    59$-54$,-                            ;
                                                                                    65$-54$,-                            ;
                                                                                    65$-54$,-                            ;
                                                                                    65$-54$,-                            ;
                                                                                    65$-54$,-                            ;
                                                                                    65$-54$,-                            ;
                                                                                    62$-54$,-                            ;
                                                                                    62$-54$,-                            ;
                                                                                    62$-54$,-                            ;
                                                                                    62$-54$                              ;
                             89      AB               5A  B0 002FD  55$:    MOVW    EL_SIZE, -119(CCB)                   ; 1103
                                                      6A  11 00301          BRB     65$                                  ;
                             89      AB               02  B0 00303  56$:    MOVW    #2, -119(CCB)                        ; 1111
                                                      64  11 00307          BRB     65$                                  ;
                                     04               5A  D1 00309  57$:    CMPL    EL_SIZE, #4                          ; 1120
                                                      06  1E 0030C          BGEQU   58$                                  ;
                             89      AB               07  B0 0030E          MOVW    #7, -119(CCB)                        ;
                                                      59  11 00312          BRB     65$                                  ;
                             89      AB               0C  B0 00314  58$:    MOVW    #12, -119(CCB)                       ;
                                                      53  11 00318          BRB     65$                                  ;
                 50                  5A               03  78 0031A  59$:    ASHL    #3, EL_SIZE, R0                      ; 1131
                                     50               02  C0 0031E          ADDL2   #2, R0                               ;
                                     50               03  C6 00321          DIVL2   #3, R0                               ;
                                                      50  D6 00324          INCL    R0                                   ;
                     0000FFFF        8F               50  D1 00326          CMPL    R0, #65535                           ;
                                                      05  15 0032D          BLEQ    60$                                  ;
                                     50      FFFF     8F  3C 0032F          MOVZWL  #65535, R0                           ;
                                     07               50  D1 00334  60$:    CMPL    R0, #7                               ;
                                                      03  18 00337          BGEQ    61$                                  ;
                                     50               07  D0 00339          MOVL    #7, R0                               ;
                             89      AB               50  B0 0033C  61$:    MOVW    R0, -119(CCB)                        ;
                                                      2B  11 00340          BRB     65$                                  ;
                                     04               5A  D1 00342  62$:    CMPL    EL_SIZE, #4                          ; 1146
                                                      1E  13 00345          BEQL    64$                                  ;
                                     08               5A  D1 00347          CMPL    EL_SIZE, #8                          ; 1153
                                                      0A  12 0034A          BNEQ    63$                                  ;
                             89      AB 02100019      8F  D0 0034C          MOVL    #34603033, -119(CCB)                 ; 1156
                                                      17  11 00354          BRB     65$                                  ; 1143
                                     10               5A  D1 00356  63$:    CMPL    EL_SIZE, #16                         ; 1160
                                                      0A  12 00359          BNEQ    64$                                  ;
                             89      AB 0321002A      8F  D0 0035B          MOVL    #52494378, -119(CCB)                 ; 1163
                                                      08  11 00363          BRB     65$                                  ; 1143
                             89      AB 0207000F      8F  D0 00365  64$:    MOVL    #34013199, -119(CCB)                 ; 1170
                                     58               14  C2 0036D  65$:    SUBL2   #20, FMT_CODE                        ; 1185
                                     5E               0C  C0 00370  66$:    ADDL2   #12, SP                              ; 1196
                                                          05 00373          RSB                                          ;

; Routine Size:  884 bytes,    Routine Base:  _FOR$CODE + 0092
; 1138          1197  1
; 1139          1198  1 END                                 ! End of module FOR$$FMT_INTRP
; 1140          1199  1
; 1141          1200  0 ELUDOM




;                                       PSECT SUMMARY
;
;       Name                        Bytes                       Attributes
;
;  _FOR$CODE                           1030  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                             Library Statistics
;
;                                               -------- Symbols --------       Pages       Processing
;       File                                    Total    Loaded   Percent       Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1             9776         0         0         581         00:01.1
;  _$255$DUA28:[FORRTL.OBJ]FORLIB.L32;1           711       211        29          52         00:00.6
;  _$255$DUA28:[FORRTL.OBJ]RTLLIB.L32;1            36         0         0           8         00:00.1




;                                         COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:FORFMTINT/OBJ=OBJ$:FORFMTINT MSRC$:FORFMTINT/UPDATE=(ENH$:FORFMTINT
;       )

; Size:          922 code + 108 data bytes
; Run Time:         00:29.5
; Elapsed Time:     01:23.3
; Lines/CPU Min:     2441
; Lexemes/CPU-Min: 23045
; Memory Used:  386 pages
; Compilation Complete
```

0180 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



FORFIND
LIS

FORFMTINT
LIS

FOREXITHA
LIS

FORENODEF
LIS

FORENCOMF
LIS

FORIDATE
LIS

FORENDFIL
LIS

FORERRSNS
LIS

FOREXIT
LIS

FORERROR
LIS

FORFMTCP
LIS

FORENCOMO
LIS

FORINIDES
LIS